

REMARKS

Claims 1, 4-8, and 10-13 are pending after entry of the present Amendment.

Applicants respectfully request reconsideration of the application in view of the following remarks submitted in support thereof.

Rejections under 35 U.S.C. § 103(a):

Claims 1, 4-8, and 10-13 were rejected under 35 U.S.C. § 103(a), as being unpatentable over Ma et al. (U.S. Patent 5,920,725) (hereinafter "Ma"), in view of Ferguson et al. (U.S. Patent 6,976,079) (hereinafter "Ferguson"), and in further view of Arnaiz et al. (U.S. Patent 7,080,371) (hereinafter "Arnaiz"). This rejection is respectfully traversed.

Ma discloses a distributed application with an adaptor 80 for updating objects cataloged by a meta server 70. Meta server 70 receives change requests from a runtime update tool 76 and modifies object class definitions. The update tool 76 provides a way for a user to inject an update. *See col. 7, lines 60-66, col. 8, lines 1-6, and Figure 3.*

As noted by Ma, these updates are not related to specific services. First, specific objects are updated in the repository 62. Then, notifications are made to the server application 86, where objects are updated. Finally, the notification is made to the client 88, where the objects can be updated. Ma's teachings are directed to a distributed system, which executes objects both on the server side and the client side. Ma does not explicitly suggest or teach generating an upgraded control module defining upgraded application-specific policies relative to the original control module or maintaining a recoverable state of the upgraded control module at a state server. In contrast, the control module of the claimed invention generates an upgraded control module defining upgraded application specific policies relative to the original control module and a recoverable state of the upgraded control module is maintained at a state server for recovery during upgrades.

Ferguson discloses a distributed computing environment wherein a first application server, on which a software to-be-upgraded resides, re-directs any new requests for the software to a second application server using a router, to service the request for the current software version while the software is being upgraded on the first application server. Upon upgrade, the first application server accepts new service requests from clients using the upgraded version of the software. *See Col. 5, lines 27-40.* If the software on the first application server is servicing an active client request, the system waits till the application server is no longer servicing the request before upgrading. *See Col. 5, lines 48-52.* The wait for upgrading of software on each application server adds to delay in running the application on that application server.

Arnaiz is not analogous art as Arnaiz deals with automatic upgrade of software components of an application on a client. A client defines an upgrade kit, writes this definition to a database on a server. The client, then, generates upgrade kit files at the client which is copied to the server. The upgrade kit files create an upgrade kit archive file when the upgrade kit files are ready to be released to the clients. (*See column 5, lines 14-23*). According to Arnaiz, at certain pre-defined times, a system program on the server verifies that software components for the currently running programs are up-to-date. (*See column 6, lines 19-45*). In order to accomplish this, the system program on the server invokes an upgrade wizard to apply an upgrade kit. The upgrade wizard then compares the version of the software components on the clients with a required version of software components in a database on the server and releases the proper upgrades found in the upgrade kit archive file on the server when the software components' versions differ from an acceptable range and informs the client to perform the upgrade. The client then upgrades the software component using the released upgrades by invoking the upgrade wizard and exits. Additionally, in order to upgrade the software components, the server sends shutdown messages to all currently

active server components, waits for all server components to stop, then, invokes the upgrade wizard and exits. (*See column 6, lines 43-45*). This means that the server and the client are not online during the software component upgrades. The original program is then re-started on the server after successful completion of the upgrade contributing to delay in making the software component available to the client and disruption in the running of the application. (*See column 6, lines 43-45, lines 64-67*). This is not the same as the claimed invention where the system continues to provide services to clients while a software upgrade is being performed. (*See claimed invention, page 12, lines 19-23*). Therefore, it would not be obvious to one skilled in the art to apply the automatic client upgrade of software components of Arnaiz (using the recoverable state of the upgrade maintained by the upgrade wizard) to the teachings of Ma and Ferguson in order to arrive at the claimed invention.

The independent claims 1 and 8 of the claimed invention describes a method wherein a control module is used in upgrading the objects. The control module supervises the life cycle of the control modules by collaborating with the runtime executive subsystem. (*See page 15, lines 12-15, page 17, lines 1- 13*). The control module of the claimed invention manages the systematic upgrade throughout the system while maintaining the application-specific strategies and policies for an application. Moreover, these upgrades are performed in place with no disruption in service. *See page 35, lines 1-2 and lines 21-23*. This feature would provide for a seamless upgrade process without causing unnecessary delay or disruption in the running of the application. Additionally, the claimed invention also provides for a failsafe upgrade by maintaining a recoverable state of the upgraded control module in a state server. (*See page 14, lines 13-16, page 19, lines 2-3, and page 27, lines 4-6*).


As can be seen from the above argument, the teachings of Ma, Ferguson and Arnaiz, combined do not provide the feature of each and every feature of the amended independent

claims 1 and 8. For at least the reasons noted herein, the Applicants respectfully submit that the now claimed invention is patentable over the cited art.

Claims 4-7 and 10-13 are directly dependent on independent claims 1 and 8 respectively. Based on arguments for independent claims 1 and 8, Applicants request the Examiner to withdraw the 35 U.S.C. §103(a) rejections on claims 4-7 and 10-13. A Notice of Allowance is respectfully requested.

If the Examiner has any questions concerning the present Amendment, the Examiner is kindly requested to contact the undersigned at (408) 774-6905. If any other fees are due in connection with filing this Amendment, the Commissioner is also authorized to charge Deposit Account No. 50-0805 (Order No. SUNMP003). A duplicate copy of the transmittal is enclosed for this purpose.

Respectfully submitted,
MARTINE PENILLA & GENCARELLA, LLP


Jayanthi Minisandram
Reg. No. 53,330

710 Lakeway Drive, Suite 200
Sunnyvale, CA 94085
Telephone: (408) 749-6900
Facsimile: (408) 749-6901
Customer No. 32291